

REFERENCES

- [1] A. Haas, A. Rossberg, D. L. Schuff, D. L. Schuff, B. L. Titzer, M. Holman, D. Gohman, L. Wagner, A. Zakai, and J. F. Bastien, “Bringing the web up to speed with webassembly,” *PLDI*, 2017.
- [2] P. Mendki, “Evaluating webassembly enabled serverless approach for edge computing,” in *2020 IEEE Cloud Summit*, pp. 161–166, 2020.
- [3] M. Jacobsson and J. Wåhslén, “Virtual machine execution for wearables based on webassembly,” in *EAI International Conference on Body Area Networks*, pp. 381–389, Springer, Cham, 2018.
- [4] Bytecode Alliance, “Bytecode Alliance.” <https://bytecodealliance.org/>, 2019.
- [5] “Webassembly system interface.” <https://github.com/WebAssembly/WASI>, 2021.
- [6] D. Bryant, “Webassembly outside the browser: A new foundation for pervasive computing,” in *Proc. of ICWE 2020*, pp. 9–12, 2020.
- [7] B. Spies and M. Mock, “An evaluation of webassembly in non-web environments,” in *2021 XLVII Latin American Computing Conference (CLEI)*, pp. 1–10, 2021.
- [8] E. Wen and G. Weber, “Wasmachine: Bring iot up to speed with a webassembly os,” in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 1–4, IEEE, 2020.
- [9] D. Lehmann, J. Kinder, and M. Pradel, “Everything old is new again: Binary security of webassembly,” in *29th USENIX Security Symposium (USENIX Security 20)*, USENIX Association, Aug. 2020.
- [10] M. Kim, H. Jang, and Y. Shin, “Avengers, assemble! survey of webassembly security solutions,” in *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*, pp. 543–553, 2022.
- [11] J. Cabrera Arteaga, O. Floros, O. Vera Perez, B. Baudry, and M. Monperrus, “Crow: code diversification for webassembly,” in *MADWeb, NDSS 2021*, 2021.
- [12] P. K. Gadepalli, S. McBride, G. Peach, L. Cherkasova, and G. Parmer, “Sledge: A serverless-first, light-weight wasm runtime for the edge,” in *Proceedings of the 21st International Middleware Conference*, p. 265–279, 2020.
- [13] R. Gurdeep Singh and C. Scholliers, “Warduino: A dynamic webassembly virtual machine for programming microcontrollers,” in *Proceedings of the 16th ACM SIGPLAN International Conference on Managed Programming*

- Languages and Runtimes*, MPLR 2019, (New York, NY, USA), pp. 27–36, ACM, 2019.
- [14] S. S. Salim, A. Nisbet, and M. Luján, “Trufflewasm: A webassembly interpreter on graalvm,” in *Proceedings of the 16th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, VEE ’20, (New York, NY, USA), p. 88–100, Association for Computing Machinery, 2020.
 - [15] E. Johnson, E. Laufer, Z. Zhao, D. Gohman, S. Narayan, S. Savage, D. Stefan, and F. Brown, “Wave: a verifiably secure webassembly sandboxing runtime,” in *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 2940–2955, 2023.
 - [16] Q. Stiévenart and C. De Roover, “Wassail: a webassembly static analysis library,” in *Fifth International Workshop on Programming Technology for the Future Web*, 2021.
 - [17] I. Bastys, M. Algehed, A. Sjösten, and A. Sabelfeld, “Secwasm: Information flow control for webassembly,” in *Static Analysis* (G. Singh and C. Urban, eds.), (Cham), pp. 74–103, Springer Nature Switzerland, 2022.
 - [18] T. Brito, P. Lopes, N. Santos, and J. F. Santos, “Wasmati: An efficient static vulnerability scanner for webassembly,” *Computers & Security*, vol. 118, p. 102745, 2022.
 - [19] F. Breitfelder, T. Roth, L. Baumgärtner, and M. Mezini, “Wasma: A static webassembly analysis framework for everyone,” in *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pp. 753–757, 2023.
 - [20] F. Marques, J. Fragoso Santos, N. Santos, and P. Adão, “Concolic execution for webassembly (artifact),” Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.
 - [21] E. Johnson, D. Thien, Y. Alhessi, S. Narayan, F. Brown, S. Lerner, T. McMullen, S. Savage, and D. Stefan, “, : Sfi safety for native-compiled wasm,” *Network and Distributed Systems Security (NDSS) Symposium*.
 - [22] W. Fu, R. Lin, and D. Inge, “Taintassembly: Taint-based information flow control tracking for webassembly,” *arXiv preprint arXiv:1802.01050*, 2018.
 - [23] D. Lehmann and M. Pradel, “Wasabi: A framework for dynamically analyzing webassembly,” in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 1045–1058, 2019.
 - [24] D. Lehmann, M. T. Torp, and M. Pradel, “Fuzzm: Finding memory bugs through binary-only instrumentation and fuzzing of webassembly,” *arXiv preprint arXiv:2110.15433*, 2021.